(51) International Patent Classification [6] : G06F 13/00, 13/14	**A1**	(11) International Publication Number: **WO 99/48007** (43) International Publication Date: 23 September 1999 (23.09.99)

(21) International Application Number: PCT/IL99/00157

(22) International Filing Date: 18 March 1999 (18.03.99)

(30) Priority Data:
09/044,167   19 March 1998 (19.03.98)   US

(71)(72) Applicants and Inventors: COHEN, Barak [IL/IL]; Shaked Street 17, 48571 Rosh Hain (IL). KARIV, Asher [IL/IL]; Shaked Street 17, 48571 Rosh Hain (IL). GLICK, Amir [IL/IL]; Shaked Street 17, 48571 Rosh Hain (IL).

(74) Agent: FRIEDMAN, Mark, M.; Beit Samueloff, Haomanim 7, 67897 Tel Aviv (IL).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: A METHOD AND SYSTEM FOR OPERATING DISTRIBUTED HARDWARE DEVICES REMOTELY ON A NETWORK ACROSS DIFFERENT PLATFORMS
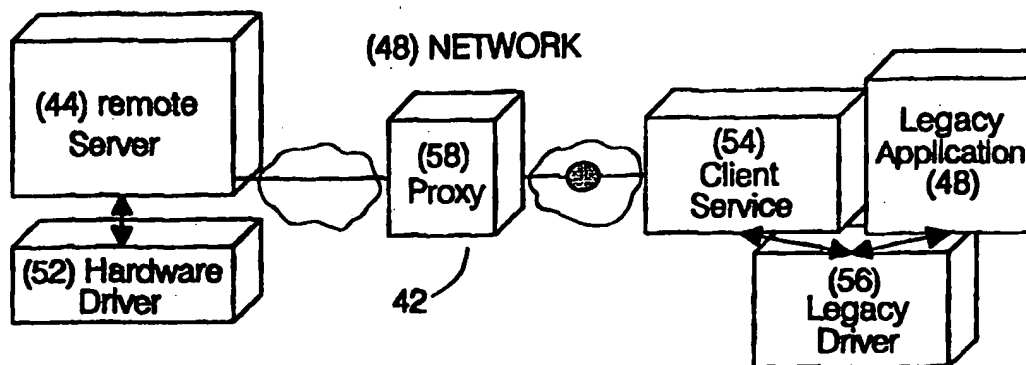
(57) Abstract

A system for remote operation of a hardware device by a client computer (54), comprising: a server computer for being connected to the hardware device, the server being connected to the client computer through a network (48); and a driver (52) for communicating with the hardware device, the driver being operated by the server computer and a remote server (44) for communicating with the driver and the client computer, the remote server being operated by the server computer, such that the client computer is able to communicate with the hardware device through the remote server and a legacy application for being operated by the client computer, such that the legacy application (48) only communicates indirectly with the remote server and a legacy driver (56) for being operated by the client computer and interacting directly with the legacy application.

A METHOD AND SYSTEM FOR OPERATING DISTRIBUTED HARDWARE DEVICES REMOTELY ON A NETWORK ACROSS DIFFERENT PLATFORMS

FIELD AND BACKGROUND OF THE INVENTION

The present invention relates to a system and method for remote operation of
5   hardware devices and, more specifically, to a system and a method which enables
distributed hardware devices to be browsed, operated and managed on a network
across platforms.

Networks are able to provide a connection for many different types of
computers being operated by different operating systems. Such networks are useful
10   for communication between computers, for example with electronic mail.
However, networks also enable distributed resources to be available to remote users.
For example, disk storage could be provided on one server computer but then
accessed by a plurality of remote client computers. Such distribution of resources
can be very useful for corporate networks, enabling two remote users to examine
15   files without requiring local copies of these files to be available, for example.

For the UNIX operating system, hardware devices can be mounted for local
operation by a remote computer. For example, a computer being operated
according to the UNIX operating system could mount a tape backup device attached
to a remote computer through the network. The local computer could then control
20   the operation of the remote tape backup device. Such functionality is particularly
useful when each computer of a group on a network has attached disk storage space
which must be backed up. Rather than purchase and install a separate tape backup
device for each computer, the group of computers being operated according to the
UNIX operating system could share one such device. Thus, the ability to share
25   hardware devices across a network is clearly desirable.

Unfortunately, one of the most popular operating systems, the Windows™
group of related operating systems for the PC (personal computer) computer as well
as other operating systems, does not enable hardware devices to be operated by
remote computers. Furthermore, there is no standard way for clients that run one
30   operating system to access hardware on another operating system. Continuing with
the example of the tape backup device, a remote computer being operated according

to one of this group of operating systems would not be able to operate the tape backup device as though the device were directly attached to the remote computer. Thus, a separate tape backup device would be required for local operation.

The current inability of available operating systems to permit sharing of
5   hardware devices across a network results in duplication of hardware resources, even for such infrequently used devices as a tape backup device. Furthermore, adding a new hardware device to a computer can also be difficult, since frequently a new serial port must be added to the computer in order to communicate with the new device. Thus, the duplication of hardware resources is highly inefficient and
10   inconvenient, particularly for organizations with large networks of many computers.

A far more convenient and efficient system would enable many different computers on a network to share remote hardware resources, such as tape backup devices, as though these devices were directly attached to the local computer. Such a system would sharply reduce or eliminate duplication of hardware resources, and
15   simplify administration and operation of these resources. In addition, such a system would enable embedded platforms, such as a smart-card device being operated according to the Windows CE™ operating system or another embedded operating system, to interact with remote computers and server as hardware servers for other computers. Unfortunately, such a system for sharing hardware resources across a
20   network is not currently available.

There is therefore a need for, and it would be useful to have, a method and a system for sharing hardware resources across a network and across operating systems, so that remote computers could share hardware devices as though these devices were directly attached to the remote computer.

25

SUMMARY OF THE INVENTION

According to the present invention, there is provided a system for remote operation of a hardware device by a client computer, comprising: (a) a server computer for being connected to the hardware device, the server computer being
30   connected to the client computer through a network; (b) a driver for communicating

with the hardware device, the driver being operated by the server computer; and (c) a remote server for communicating with the driver and the client computer, the remote server being operated by the server computer, such that the client computer is able to communicate with the hardware device through the remote server.

5   Preferably, the system further includes a plurality of client computers, wherein the remote server is able to cache data from the hardware device and to provide the data to each of the plurality of client computers individually, such that substantially all of the plurality of client computers are able to receive substantially the same data. Also preferably, the client computer is operated according to a 32-bit

10   version of a Windows™ operating system. More preferably, the operating system for the client computer is each individually selected from the group consisting of Windows95™, Windows CE™, Windows NT™ and Windows98™. The server computer is preferably operated by a multithreaded operating system that supports TCP/IP. Alternatively and preferably, one of the server computer and the client

15   computer is operated according to a Java-VM operating system.

According to preferred embodiments of the present invention, the remote server and the client computer are capable of communication according to the TCP/IP suite of protocols..

According to other preferred embodiments of the present invention, the

20   remote server and the client computer communicate according to different communication protocols, and the system further includes: (d) a proxy module for translating between the remote server and the client computer, such that the remote server and the client computer are able to communicate. Preferably, the proxy module further transforms a data format for the remote server and the client

25   computer, such that the remote server and the client computer are able to exchange data.

According to still other preferred embodiments of the present invention, the system further includes (d) a legacy application for being operated by the client computer, such that the legacy application only communicates indirectly with the

30   remote server; (e) a legacy driver for being operated by the client computer and for

interacting directly with the legacy application; and (f) a client service module for being operated by the client computer and for communicating with the legacy driver, the client service module communicating with the remote server, such that the legacy application is able to interact with the hardware device through the

5 legacy driver, the client service module and the remote server.

Preferably, the remote server and the client service module communicate according to different communication protocols, the system further comprising: (g) a proxy module for translating between the remote server and the client service module, such that the remote server and the client service module are able to

10 communicate. More preferably, the proxy module transforms a data format between the legacy application and the remote server, such that the legacy application and the remote server are able to exchange data.

Alternatively and preferably, the remote server and the client service module communicate according to substantially the same communication protocol.

15 According to still other preferred embodiments of the present invention, the system further includes (g) a virtual bus driver, the virtual bus driver enabling the legacy driver and the client service module to communicate; and (h) a remote class module, the remote class module registering the hardware device with the remote server. Preferably, the client computer and the server computer are being operated

20 according to an OnNow enabled operating system.

According to another embodiment of the present invention, there is provided a system for management of a plurality of remote hardware devices, the system comprising: (a) a plurality of server computers, each of the server computers being connected to at least one of the plurality of remote hardware devices; (b) a plurality

25 of server modules, each of the plurality of server modules being run by one of the server computers; (c) a plurality of client computers, each of the client computers interacting with the server computers through the server modules; (d) a plurality of management modules for managing an interaction of the client computers and the server modules; and (e) a name server for registering information regarding the

30 server modules and providing the information to the management modules.

5

Preferably, the information includes a location of a server computer upon receipt of a browsing request. Also preferably, the information includes at least one attribute of the hardware devices. More preferably, the information includes a description of the hardware devices.

5   Hereinafter, the term "network" refers to a connection between any two computers which permits the transmission of data. Hereinafter, the term "computer" includes, but is not limited to, personal computers (PC) having an operating system such as DOS, Windows™, OS/2™ or; Macintosh™ computers; computers having JAVA™-OS as the operating system; and graphical workstations such as the

10   computers of Sun Microsystems ™ and Silicon Graphics™, and other computers having some version of the UNIX operating system such as AIX or SOLARIS™ of Sun Microsystems™; or any other known and available operating system. Hereinafter, the term "Windows™" includes but is not limited to Windows95™, Windows 3.x™ in which "x" is an integer such as "1", Windows NT™,

15   Windows98™, Windows CE™ and any upgraded versions of these operating systems by Microsoft Inc. (Seattle, Washington, USA). Hereinafter, the terms "PC" or "PC computer" refer to computers having an operating system such as DOS, Windows™, OS/2™ or Linux. Hereinafter, the term "netPC" includes, but is not limited to, a computer having JAVA™-OS or Java-VM as the operating system.

20   Hereinafter, the terms "Embedded platform" refer to computers having an embedded operating system such as "Windows CE", pSOS, QNX or any other embedded operating system.

For the present invention, a software application could be written in substantially any suitable programming language, which could easily be selected by

25   one of ordinary skill in the art. The programming language chosen should be compatible with the computing platform according to which the software application is executed. Examples of suitable programming, languages include, but are not limited to, C, C++ and Java.

In addition, the present invention could be implemented as software,

30   firmware, or hardware or as a combination thereof. For any of these

6

implementations, the functional steps performed by the method could be described as a plurality of instructions performed by a data processor.

BRIEF DESCRIPTION OF THE DRAWINGS

5   The foregoing and other objects, aspects and advantages will be better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, wherein:

FIG. 1 is a schematic illustration of a prior art system for operating a hardware device;

10   FIG. 2 is a schematic illustration of a native system for remote operation of hardware according to the present invention;

FIG. 3 is a schematic illustration of the system of Figure 2 with a proxy according to the present invention;

FIG. 4 is a schematic illustration of a legacy system for remote operation of

15   hardware according to the present invention;

FIG. 5 is a schematic illustration of the system of Figure 4 with a proxy according to the present invention;

FIG. 6 is a schematic illustration of a virtual bus system for remote operation of hardware according to the present invention; and

20   FIG. 7 is a schematic illustration of a management system for remote operation of hardware according to the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention is of a method and a system to administer, browse and

25   operate a remote hardware device over a network by a local computer as though the remote hardware device is directly attached to the local computer.

The principles and operation of a method and a system for remote operation of a hardware device according to the present invention may be better understood with reference to the drawings and the accompanying description, it being

30   understood that these drawings are given for illustrative purposes only and are not

meant to be limiting.

Although the following description will center upon the implementation of the present invention with the Windows™ operating systems and various features of these operating systems, it is understood that this is for the purposes of discussion only and is not meant to be limiting in any way. In particular, the present invention is intended to be adaptable to many different operating systems, as well as for cross-platform operation, such that computers being operated according to different operating systems are able to interact according to the present invention.

Referring now to the drawings, Figure 1 depicts the currently available prior art system for operation of a hardware device. A prior art system 10 features a local computer 12, which could be a PC or netPC computer, for example. Local computer 12 has a hardware device 14 directly attached. Hardware device 14 could be a tape backup device, for example. The operation of hardware device 14 is controlled by a software application 16, which communicates with hardware device 14 through a driver 18. Software application 16, driver 18 and hardware device 14 are all operated through local computer 12.   Local computer 12 controls the operation of hardware device 14, such that any communication with or instructions to hardware device 14 can only be given by local computer 12. Thus, even if local computer 12 was connected to another, remote computer through a network (not shown), the remote computer would still not be able to control the operation of hardware device 14, which is clearly a disadvantage of prior art system 10.

Figure 2 is a schematic illustration of a system for remote operation of hardware according to the present invention, for a native client. Figure 2 shows a remote system 20 for a native client 22. Remote system 20 features a remote server 24, which is a software module being operated by a server computer (not shown). The server computer is connected to native client 22 through a network 26. For this embodiment of the present invention, both native client 22 and the server computer are running the same communication protocol, such as TCP/IP(or other socket based communication protocol). Native client 22 is a computer which uses the same communication protocol as the server computer. The server computer could

8

be a computer being operated according to the Windows NT™ operating system, for example, or any other suitable operating system. Native client 22 could also be a computer being operated according to the Windows NT™ operating system, for example or any other suitable operating system.

5   Remote system 20 also features a hardware driver 28 for operating a hardware device which is attached to the server computer (not shown). Remote server computer 24 is able to interact with hardware driver 28 in order to enable the hardware device to be exported through network 26. Remote server 24 supports both point to point connections, as shown, and a point to multipoint connection.

10   The point to multipoint connection enables the exported hardware device to be shared by multiple native clients 22.

Remote server 24 can also provide multicasting streaming services by caching data from hardware driver 52, and then storing the cached data. Remote server 24 then serves the cached data upon request by native clients 22. Such

15   streaming of the data enables all of the native clients 22 to get the same information.

In any case, native client 22 is also able to control the operation of the exported hardware device by directly interacting with hardware driver 28, as though the hardware device was directly attached to native client 22. Such interaction occurs because the hardware device is exported by remote server 24, which

20   provides the support for the direct interaction of native client 22 and the exported hardware device. Since both the operating system of native client 22 and the operating system of the computer running remote server 24 share the same communication model and since an object communication model on the client, such as DCOM (distributed component object model), exports the hardware interfaces,

25   the interaction of remote server 24 and software applications running on native client 22 occurs through this object communication model. Thus, no additional software is required for native client 22 to run in order for native client 22 to control the hardware device.

DCOM is an object protocol which enables software objects to communicate

30   directly across a network, as though both objects were being operated by the same

local computer. In other words, each software object is able to call the methods of the other software object directly over the network. DCOM enables any software application running on native client 22 to communicate directly with remote server 24 in order to communicate with, and to send instructions to, the hardware device.

5   Of course, DCOM is simply one example of an object communication model, and is not meant to be limiting in any way.

Figure 3 shows a proxy remote system 30 according to the present invention. In this embodiment, proxy remote system 30 also features a remote server 32 and a native client 34, similar to Figure 2. Remote server 32 is also a

10   software module being operated by a server computer (not shown). The server computer is connected to a hardware device (not shown) which is controlled through a hardware driver 36. The server computer is connected to native client 34 through a network 38. For this embodiment of the present invention, native client 34 and the server computer are not running the same object communication

15   protocol, and could be different platforms.

In order for native client 34 to communicate with remote server 32, a proxy 40 is provided. Proxy 40 is a software module running on a computer (not shown) which is connected to network 38. Proxy 40 is able to translate the communication protocol used by remote server 32 into the protocol which is used

20   by native client 34. If necessary, proxy 40 is able to transform the data from the format used by remote server 32 to the format used by native client 34, and vice versa. For example, the Windows CE™ operating system does not support DCOM. If remote server 32 were being operated according to the Windows CE™ operating system, proxy 40 would translate the DCOM-based communication from native

25   client 34 into TCP/IP communication for remote server 32. Proxy 40 could be used to translate many other types of communication protocols, for example in order to further promote cross-platform compatibility. Thus, remote server 32 could communicate with many different types of native clients 34 through proxy 40.

Figure 4 shows a system according to the present invention for enabling

30   interactions of a hardware device with a legacy client. A legacy system 42 is

shown, again with a remote server **44** connected to a legacy client **46** through a network **48**. Legacy client **46** is a computer which runs a legacy application **50**, which can be software or a driver. Legacy application **50** interacts with a hardware device controlled through a hardware driver **52** in a transparent manner. Hardware

5   driver **52** is run on the computer which is running remote server **44** (not shown). The hardware device is also connected to the same computer (not shown).

In order for legacy application **50** to interact with the hardware device, the hardware device must appear to be connected directly to legacy client **46** as far as legacy application **50** is concerned. Therefore, all commands and instructions to,

10 and communication with, the hardware device must appear to occur locally. In order for such communication to occur locally, legacy client **46** also runs two other software modules: a client service module **54** and a legacy driver **56**. Client service module **54** interacts with remote server **44** in order to communicate with the hardware device through hardware driver **52**. For the purposes of the present

15 invention, client service module **54** is an example of software operated by a native client, and can use native client communication protocols to communicate with remote server **54**.

Legacy driver **56** is a software module which also runs on legacy client **46** and emulates the behavior of the hardware device as though the hardware device

20 was installed on legacy client **46**. Legacy driver **56** is specially adapted for each operating system, so that legacy application **50** can communicate with legacy driver **56** as though legacy driver **56** was the hardware driver for the remote hardware device. Legacy driver **56** understands how to communicate with legacy application **50** through a hardware profile. The hardware profile is a data profile on legacy

25 client **46** which indicates the protocol to be used when communicating with legacy application **50** as well as the necessary configuration parameters, so that legacy driver **56** can expose itself to the operating system of legacy client **46** as the proper type of hardware device.

Legacy driver **56** then communicates with client service module **54**, which

30 communicates in turn with remote server **44**. Remote server **44** then communicates

11

with the hardware device through hardware driver **52**. Thus, legacy system **42** enables legacy application **50** to interact with the hardware device as though the hardware device was installed on the local computer, which is legacy client **46**.

   Figure 5 shows a second embodiment of legacy system **42**, in which remote
5 server **44** communicates with legacy client **46** through a proxy **58**. As for Figure 3, the computer running remote server **44** and legacy client **46** have different communication and data protocols, and could optionally be different platforms. Proxy **58** enables legacy client **46** to communicate with remote server **44**. The remaining aspects of the system are similar to that shown in Figure 4.

10   Figure 6 shows a virtual bus system **60** according to the present invention. Virtual bus system **60** features a remote server **62** being run on a computer (not shown). Remote server **62** is connected to a legacy client **64** through a client service module **68**. Legacy client **64** is a computer which runs client service module **68**. Legacy client **64** also features a legacy application **66** and a legacy
15 driver **70**. Legacy application **66** communicates with legacy driver **70**, while client service module **68** communicates with remote server **62**. However, legacy driver **70** communicates with client service module **68** through a virtual bus driver **72**. Virtual bus driver **72** is a software module which runs on OnNow™ enabled operating systems, such as Windows NT™ and Windows95™, or an upgraded
20 version of one of these systems such as Windows98™. OnNow™ is a standard for operating systems for controlling the operation of hardware devices. In particular, OnNow™ permits notification of virtual bus driver **72** when the hardware device controlled by hardware driver **74** becomes operational, for example by being installed on the hardware bus on the remote computer. Through the OnNow™
25 design standard, virtual bus driver **72** enables legacy driver **70** to reflect the operational status of the remote hardware device. In effect, virtual bus driver **72** simulates a bus to which hardware devices are attached. These hardware devices are exported by remote server **62** through network **48**.

   Similarly, remote server **62** communicates with a hardware driver **74** through
30 a remote class module **76**. Remote class module **76** is a software module which

runs on OnNow enabled operating systems, such as Windows NT™ and

Windows95™, or an upgraded version of one of these systems such as

Windows98™, on the same computer as remote server **62**. Remote class module **76**

provides registration services for hardware drivers **74** to be exported through

5   network **48**.

Remote class module **76** and virtual bus driver **72** (through client service

module **68**) can communicate through the Windows™ Driver Model (WDM), for

example. WDM is a bus driver which provides certain services accessible to any

software program being run by a computer operated according to one of these

10   operating systems. In particular, WDM enables a driver to receive notification as

soon as a device is connected to a local hardware bus.

Now legacy application **66** can communicate with the remote hardware

device as follows. The remote hardware device becomes activated, for example

after "sleeping". Hardware driver **74** now notifies remote class module **76**. Remote

15   class module **76** then notifies remote server **62**, which in turn notifies client service

module **68**. Client service module **68** notifies virtual bus driver **72** as though virtual

bus driver **72** was an actual hardware bus. Virtual bus driver **72** communicates with

legacy driver **70** so that legacy driver **70** can reflect the current status of the remote

hardware device. This communication occurs as though the remote hardware

20   device was connected to a local hardware bus on legacy client **64**.

Figure 7 shows a management system according to the present invention. A

management system **78** includes a plurality of remote servers **80** connected to

network **48**, which could be remote servers from any of the preceding Figures 2-6.

Similarly, management system **78** includes a plurality of clients **82** connected to

25   network **48**, which could be clients from any of the preceding Figures 2-6.

Management system **78** additionally features a plurality of remote management

modules **84** connected to network **48**. Remote management modules **84** provides

device browsing capabilities so that clients **82** can browse network **48** for hardware

devices (not shown) exported by remote servers **80**.   In addition, remote

30   management modules **84** provides device security management, so an administrator

could decide which devices an individual user could access, for example.

Management system **78** also features a name server **86**. Name server **86** contains the information from all remote servers **80** attached to network **48**. For example, name server **86** contains the information concerning the location of

5   hardware devices on network **48**, and the attributes and description of these devices (not shown). When a user sends a request for browsing to management system **78**, management system **78** then draws this information from name server **86**. Any remote server **80** which exports hardware devices preferably registers with name server **86** to indicate the location of the hardware device, and the attributes and

10   description of the hardware device.

As another example of management system **78**, management system **78** is optionally integrated with the standard directory services offered by the operating system according to the ActiveDirectory™ implementation on certain 32-bit Windows™ operating systems, such as Windows NT™ and Windows98™. This

15   integration enables management system **78** to permit the user to browse and explore devices without remote server **80** or management module **84**, through standard directory services. The hardware device is published to the ActiveDirectory™ services or other standard directory services, so that every client **82** can browse through these published device resources.

20

It will be appreciated that the above descriptions are intended only to serve as examples, and that many other embodiments are possible within the spirit and the scope of the present invention.

WHAT IS CLAIMED:

1.   A system for remote operation of a hardware device by a client computer, comprising:

    (a)   a server computer for connecting to the hardware device, said server computer connecting to the client computer through a network;

    (b)   a driver for communicating with the hardware device, said driver being operated by said server computer; and

    (c)   a remote server for communicating with said driver and the client computer, said remote server being operated by said server computer, such that the client computer is able to communicate with the hardware device through said remote server.

2.   The system of claim 1, further comprising a plurality of client computers, wherein said remote server is able to cache data from the hardware device and to provide said data to each of said plurality of client computers individually, such that substantially all of said plurality of client computers are able to receive substantially the same data.

3.   The system of claim 1, wherein said server computer and the client computer are each operated according to a 32-bit version of a Windows™ operating system.

4.   The system of claim 3, wherein said operating system for said server computer and the client computer is each individually selected from the group consisting of Windows95™, Windows CE™, Windows NT™ and Windows98™.

5.   The system of claim 1, wherein one of said server computer and the client computer is operated according to a Java-VM operating system.

6.   The system of claim 1, wherein said remote server and the client computer are capable of communication according to substantially the same object communication protocol.

7.   The system of claim 6, wherein said object communication protocol is a DCOM architecture.

8.   The system of claim 1, wherein said remote server and the client computer communicate according to different communication protocols, the system further comprising:

   (d)   a proxy module for translating between said remote server and the client computer, such that said remote server and the client computer are able to communicate.

9.   The system of claim 8, wherein said proxy module further transforms a data format for said remote server and the client computer, such that said remote server and the client computer are able to exchange data.

10.   The system of claim 1, further comprising:

   (d)   a legacy application for being operated by the client computer, such that said legacy application only communicates indirectly with said remote server;

   (e)   a legacy driver for being operated by the client computer and for interacting directly with said legacy application; and

   (f)   a client service module for being operated by the client computer and for communicating with said legacy driver, said client service module communicating with said remote server, such that said legacy application is able to interact with the hardware device through said legacy driver, said client service module and said remote server.

16

11.   The system of claim 10, wherein said remote server and said client service module communicate according to different communication protocols, the system further comprising:

(g)   a proxy module for translating between said remote server and said client service module, such that said remote server and said client service module are able to communicate.

12.   The system of claim 11, wherein said proxy module transforms a data format between said legacy application and said remote server, such that said legacy application and said remote server are able to exchange data.

13.   The system of claim 10, wherein said remote server and said client service module communicate according to substantially the same communication protocol.

14.   The system of claim 10, further comprising:

(g)   a virtual bus driver, said virtual bus driver enabling said legacy driver and said client service module to communicate; and

(h)   a remote class module, said remote class module registering the hardware device with said remote server.

15.   The system of claim 14, wherein the client computer and said server computer are being operated according to an OnNow enabled operating system.

16.   A system for management of a plurality of remote hardware devices, the system comprising:

(a)   a plurality of server computers, each of said server computers being connected to at least one of the plurality of remote hardware devices;

(b)   a plurality of server modules, each of said plurality of server modules being run by one of said server computers;

(c)   a plurality of client computers, each of said client computers interacting
        with said server computers through said server modules;

(d)   a plurality of management modules for managing an interaction of said
        client computers and said server modules; and

(e)   a name server for registering information regarding said server modules
        and providing said information to said management modules.

17.   The system of claim 16, wherein said information includes a location of a
server computer upon receipt of a browsing request.

18.   The system of claim 17, wherein said information includes at least one
attribute of said hardware devices.

19.   The system of claim 18, wherein said information includes a description
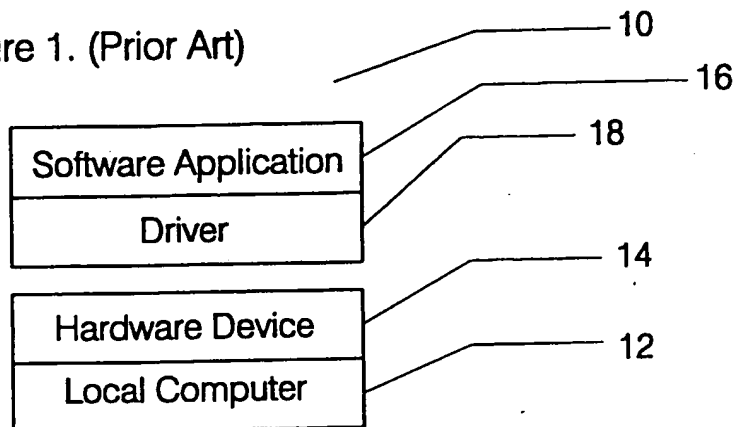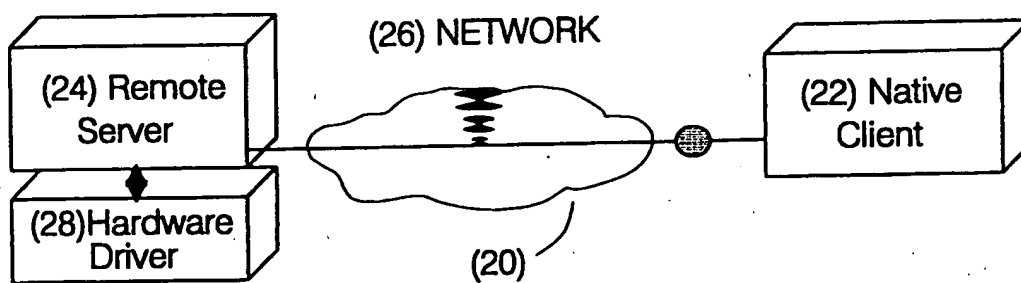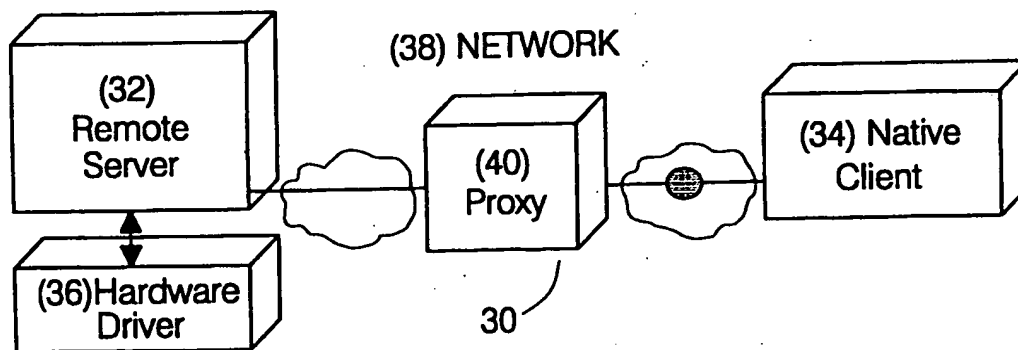of said hardware devices.

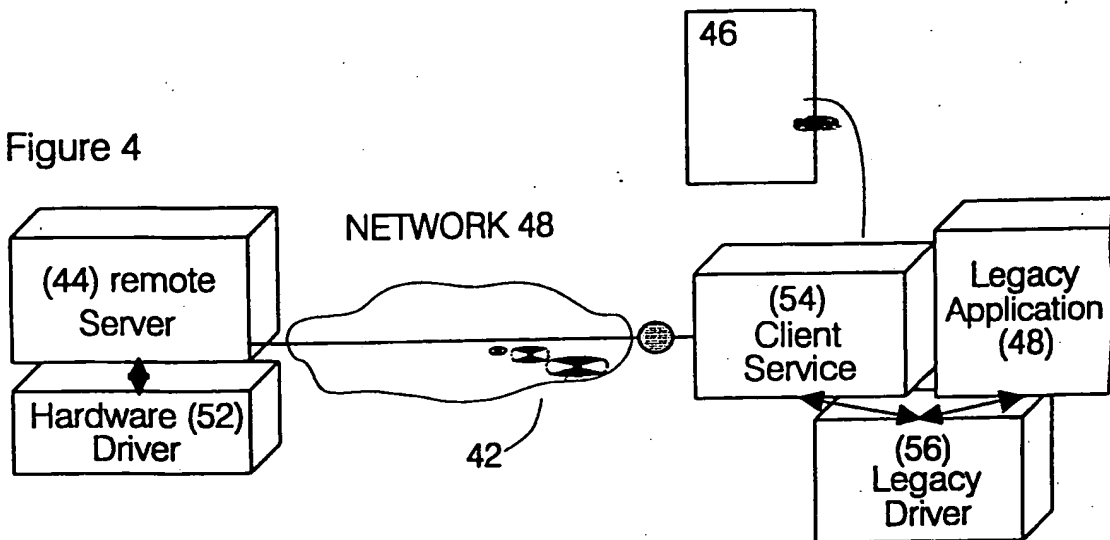Figure 1. (Prior Art)
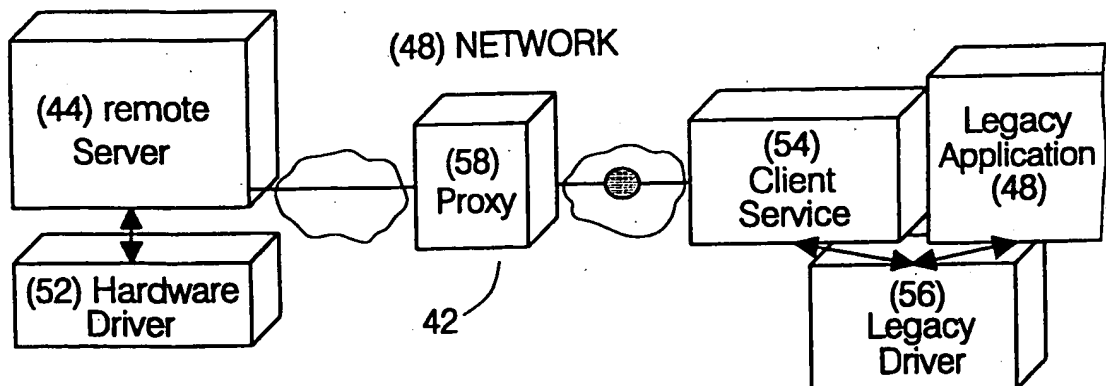

Figure 2

Figure 3

Figure 4
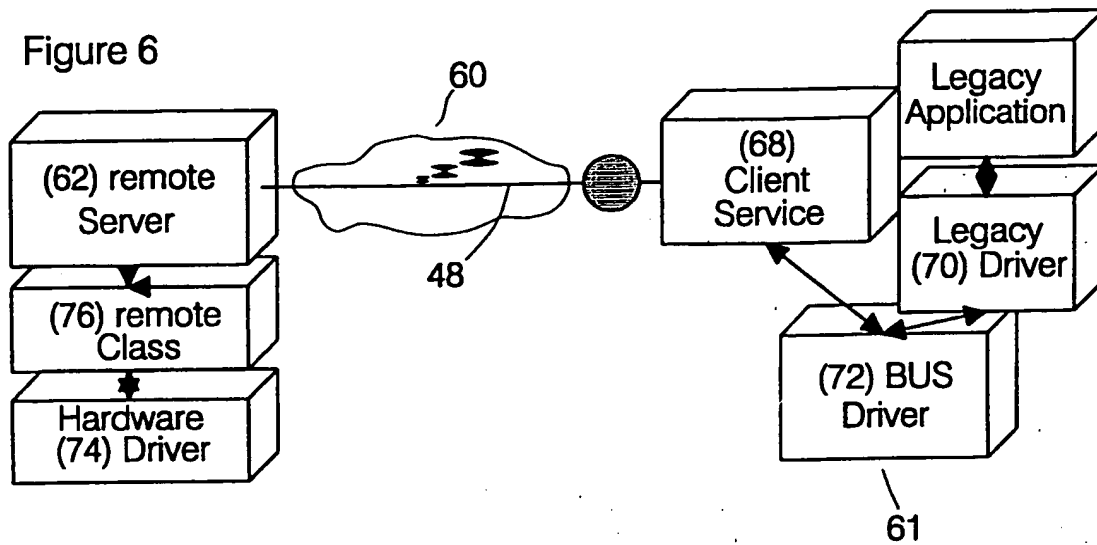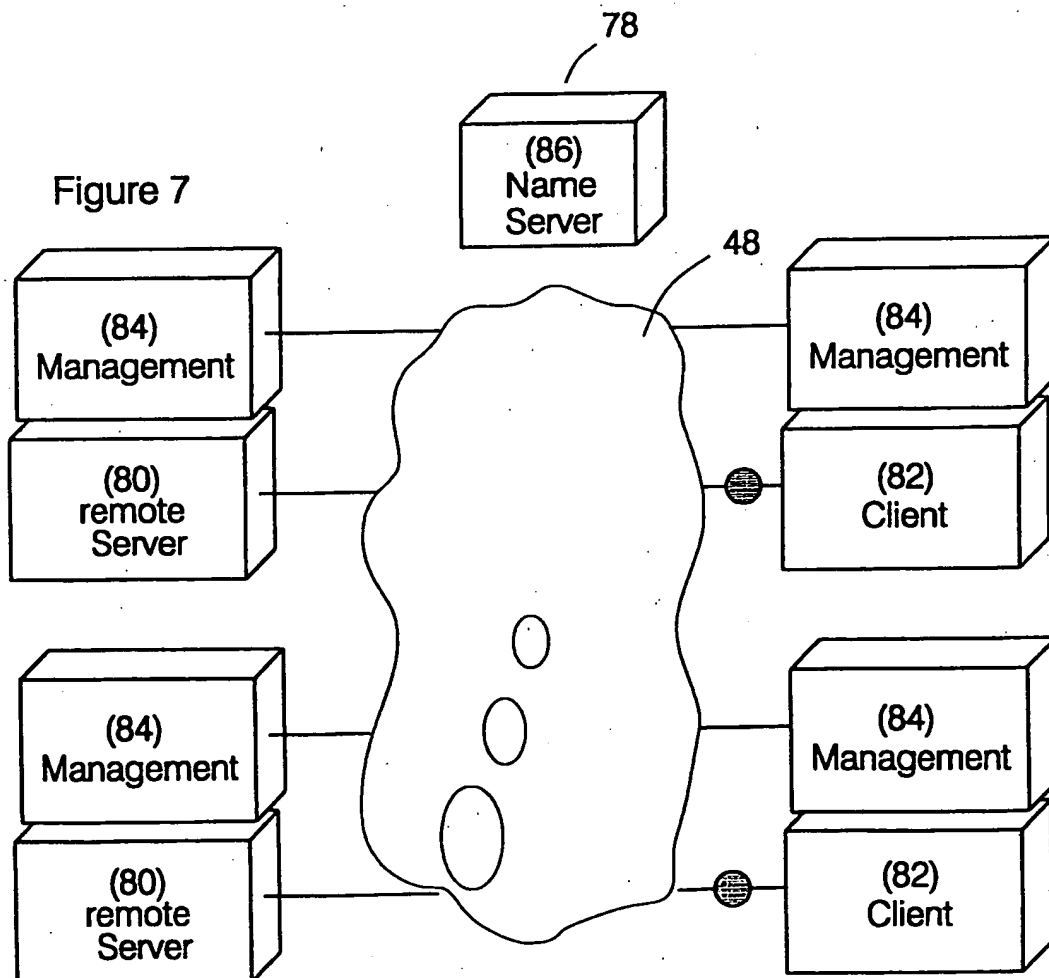

Figure 5

Figure 6

Figure 7

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IL99/00157

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 13/00, 13/14
US CL :709/203, 217, 219; 395/183.09

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. 709/203, 217, 219; 395/183.09

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, INTERNET

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 5,838,916 A (DOMENIKOS et al.) 17 NOVEMBER 1998, Col. 3, lines 1-50, Col. 6, lines 32-40, Col. 8, lines 40-60, Col. 10, lines 39-53 and Col. 13, lines 47-55 and Col. 18, lines 47-50.	1-4, 6, and 9
X, P	US 5,754,830 A (BUTTS et al.) 19 MAY 1998, Figures 1 and 2, Col. 2, lines 1-67, Col. 3, lines 1-65, Col. 4, lines 1-56 and Col. 5, lines 17-30.	5 and 10
X, P	US 5,881,230 A (CHRISTENSEN et al.) 09 MARCH 1999, Figure 8, items136-156, Col. 1, lines 60-67, Col. 2, lines 1-10, Col. 2, lines 50-65Col. 7, lines 5-67, Col. 9, lines 10-53 and Col. 12, lines 43-55	7, 8, and 11-13

X	Further documents are listed in the continuation of Box C.		See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 JULY 1999	**27 AUG 1999**

Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer FRANK J. AST*James R. Matthews* Telephone No. (703) 305-3817

Form PCT/ISA/210 (second sheet)(July 1992)*

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, P	US 5,748,980 A (LIPE et al.) 05 MAY 1998, Col. 395, lines 14-35, Col. 396, lines 48-67, Col. 397, lines 1-10, Col. 398, lines 1-67, Col. 401, lines 55-67 and Col. 402, lines 1-8, 26-52	14-19
A, E	US 5,898,835 A (TRUONG) 27 APRIL 1999, ABSTRACT, Col. 5, lines 29-67 and Col. 6, lines 1-67 and Col. 1-32	1
A, P	US 5,848,236 A (DEARTH et al.) 08 DECEMBER 1998, ABSTRACT	1, 14 AND 15
A	US 4,751,635 A (KRET) 14 JUNE 1988, ABSTRACT	1
A,E	US 5,909,545 A (FRESE, II et al.), 01 JUNE 1999, ABSTRACT	1
A	IEEE COMPUTER SOCIETY, "P1394a, DRAFT STANDARD FOR A HIGH PERFORMANCE SERIAL BUS (SUPPLEMENT)"	14